

(12) UK Patent Application (19) GB (11) 2 306 870 (13) A

(43) Date of A Publication 07.05.1997

(21) Application No 9622279.9

(22) Date of Filing 25.10.1996

(30) Priority Data

(31) 2161507

(32) 26.10.1995

(33) CA

(71) Applicant(s)

Mitel Corporation

(Incorporated in Canada - Ontario)

P O Box 13089, Kanata, Ontario K2K 1X3, Canada

(72) Inventor(s)

Deborah L Pinard

(74) Agent and/or Address for Service

John Orchard & Co

Staple Inn Buildings North, High Holborn, LONDON,
WC1V 7PZ, United Kingdom

(51) INT CL⁶

G06F 3/037

(52) UK CL (Edition O)

H4T TBLX

G4A APX

(56) Documents Cited

EP 0325885 A2

US 5422993 A

(58) Field of Search

UK CL (Edition O) H4T TBLA TBLC TBLM TBLX

INT CL⁶ G06F 3/023 3/033 3/037

ONLINE: WPI; COMPUTER

(54) COMPUTER OPERATION METHOD

(57) A single toolbar application program is used, for displaying a toolbar on a computer display. This toolbar is disposed separate from displays created by the other application programs, and contains icons representing each of the application programs. None of the other application programs have toolbars which display icons that are linked to other application programs. Selection of an icon from the single toolbar causes operation of an application program corresponding to the selected icon. Multiple icons can be selected, and multiple application programs become operational as a result. In the event a new version of an existing application program is loaded into the computer in place of the existing program, only the toolbar program is affected.

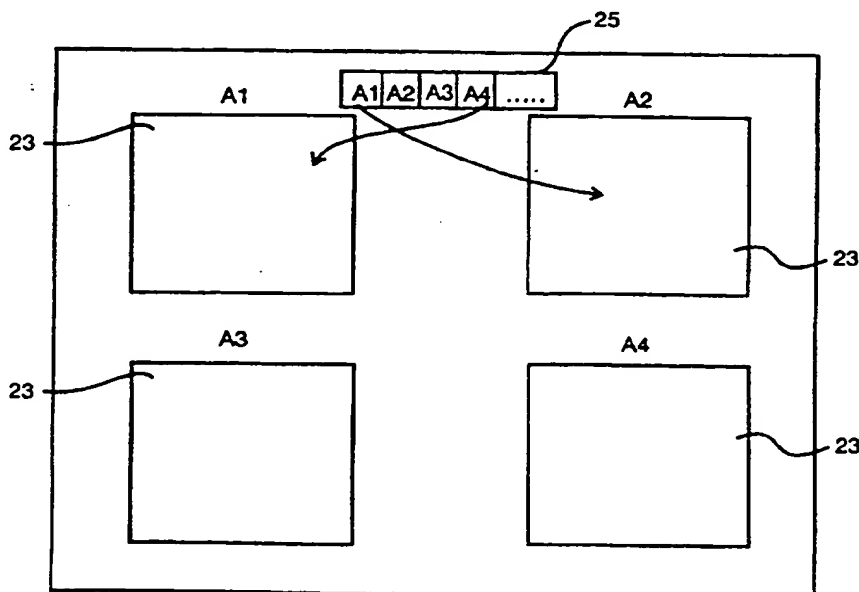


FIG. 3

GB 2 306 870 A

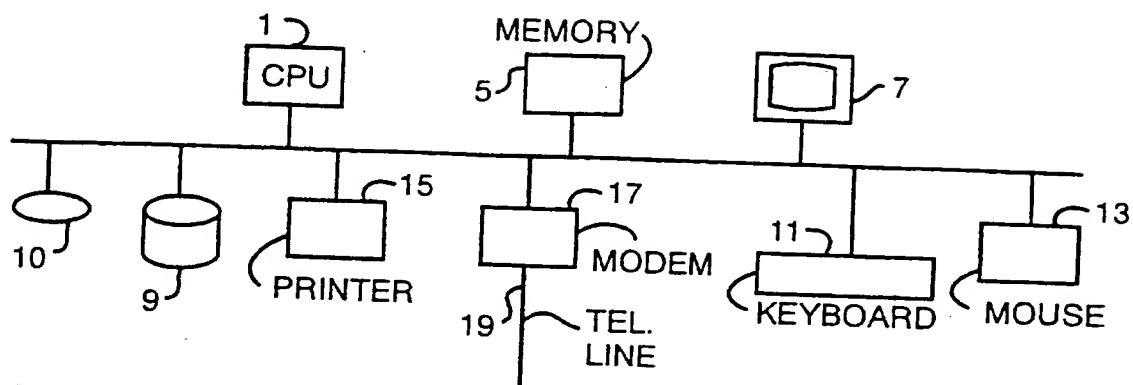


FIG. 1
PRIOR ART

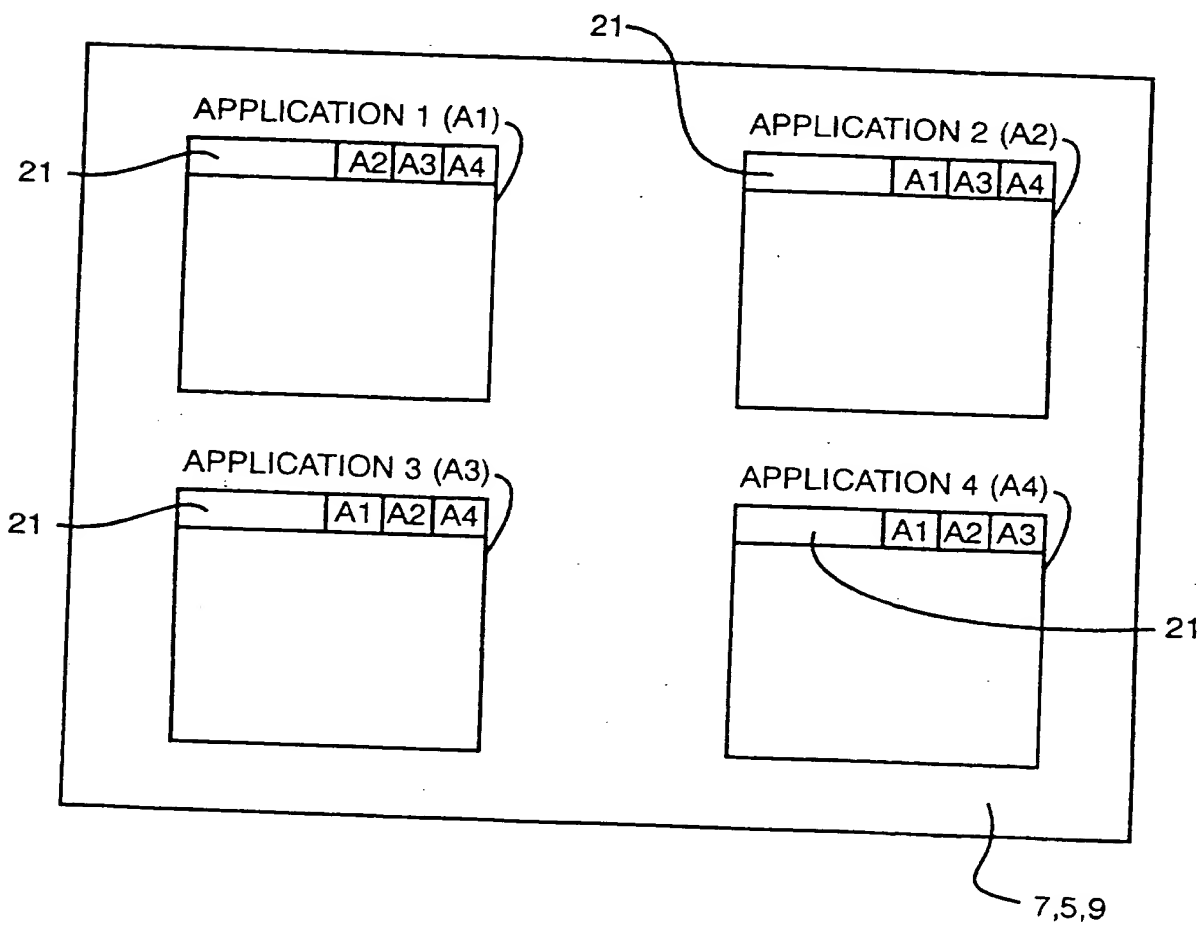


FIG. 2
PRIOR ART

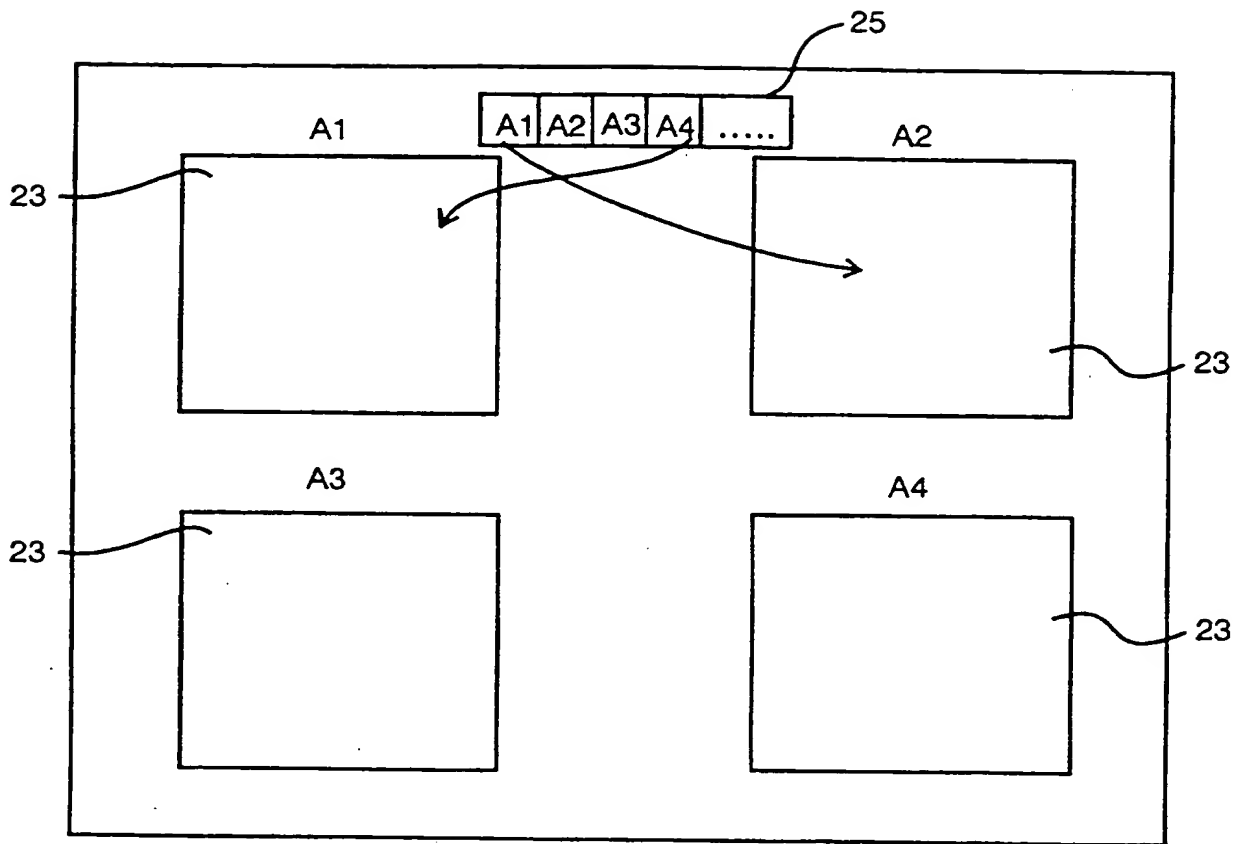


FIG. 3

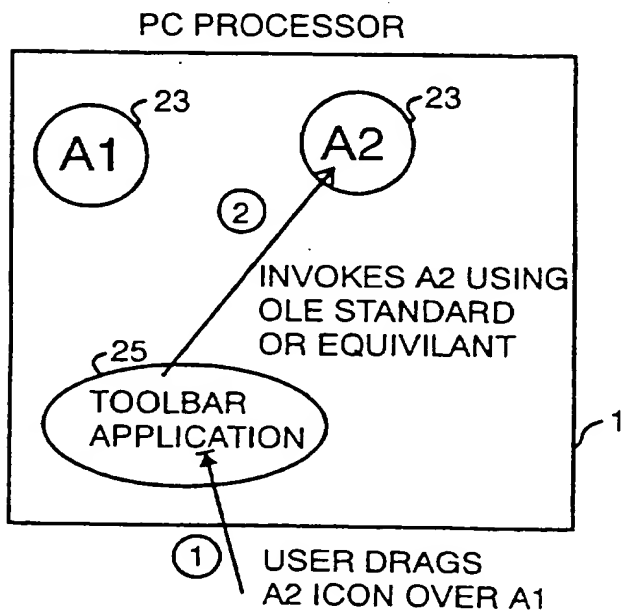


FIG. 4

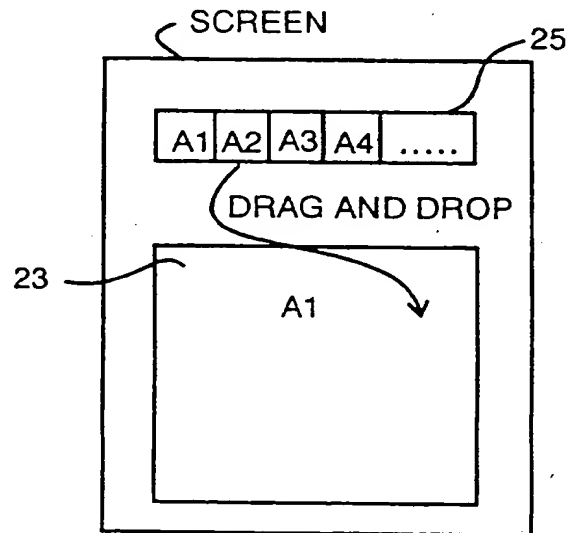


FIG. 4A

COMPUTER OPERATION METHOD

This invention relates to a method of operation of a computer. A particular method of invoking software application programs which can be linked will be described below, by way of example, in illustration of
5 the invention.

Many computer application programs are written in accordance with standards that allow other application programs to be invoked from within one which is running. For example, one such standard is the Microsoft OLE (object
10 linking) standard. This is typically carried out by providing an icon on a toolbar within one application program that represents another application program.

A problem with this linking technique is that each program must have a pointer to every other program, in order
15 that selection of an icon representing another program will invoke that other program. When a new application program is added to the group, then all of the existing application programs must change, to add a pointer to the new program, and to add an icon representing the new
20 program. This can cause errors, which can be compounded if the programs are large and have many pointers to the new program, and if there are many programs in the linked group. Further, each program must contain code to create the toolbar and the corresponding links to each of the programs
25 of the group.

In a particular embodiment to be described below by way of example in illustration of the present invention, a single toolbar application program is used, for
30 displaying a toolbar on a computer display. This toolbar is displayed separate from displays created by the other application programs, and contains icons representing each of the application programs. None of the other application programs have toolbars which display icons that are linked to other application programs.
35

Selection of an icon from the single toolbar causes operation of an application program corresponding to the

selected icon. Multiple icons can be selected, and multiple application programs become operational as a result.

In the event a new version of an existing application program is loaded into the computer in place of the existing program, only the toolbar program is affected. For example, an updated icon may be generated in place of the existing icon. It is not necessary to update the links and the icons in each of plural application programs, as was necessary in the prior art. As a result, the potential for introducing errors when loading a new or updated application program is significantly reduced.

The method of invoking software application programs to be described below, by way of example, includes the steps of storing, in a memory, a toolbar application program for displaying permanently in the foreground and controlling a toolbar on a computer display, separate from other application programs, storing in a memory other application programs which are devoid of any link to any other one of the application programs. operating the toolbar application program and displaying a toolbar comprising at least one icon over a display of the other applications, and invoking a toolbar function to cause operation of at least one of the other application programs. with parameters linked to an application program already in operation.

A better understanding of the invention will be obtained by reading the following description with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of a known personal computer,

Figure 2 is a diagram of application programs in a memory of, and as displayed in, the known personal computer,

Figure 3 is a diagram of application programs in a memory of, and as displayed in, a personal computer, in illustration of the present invention, and

Figure 4 is a block diagram for use in illustrating the operation of a computer in illustration of the present invention.

Figure 1 illustrates a personal computer on which the present invention can be implemented. The computer includes a microprocessor (CPU) 1 connected to an expansion bus 3. A random access memory (RAM) 5, a computer display 7, a hard disk drive 9, a floppy disk drive 10, and a keyboard 11 are connected to bus 3. In a typical system, a printer 15 and a modem 17 to which a telephone line 19 is connected, are also connected to bus 3. A system can have other configurations, such as including an additional high speed bus, a graphics circuit, etc.

The structure, operation, and programming of such a computer is well known, and is described for example in the text "Fundamentals Of Interactive Computer Graphics", by J.D. Foley and A. Van Dam, published by Addison-Westley Publishing Company, Inc. copyright 1982, and other publications.

It will be understood that application programs are typically stored in the hard drive 9, and upon being invoked, are at least partly brought into RAM 5. When an application program is invoked, it typically creates graphics, text, etc., which is displayed on the computer display 7 under control of the microprocessor 1. The computer operating system typically allows plural application programs to be invoked, operation of which can be time shared either together or based on which application program is currently selected. In accordance with the sharing protocols such as the aforementioned OLE, one program can be invoked from within another, whereby parameters can be shared.

For example, a word processing program can be invoked from within a spreadsheet program. Part of the spreadsheet data can be cut and pasted or otherwise moved to the word processing file. In accordance with the linking protocol, data, and changes to the data in the spreadsheet,

whether entered on the spreadsheet or the word processing file, are reflected as data entered or changed on the other of the word processing file or spreadsheet. The spreadsheet program and word processing program are thus said to be
5 linked in a group.

Figure 2 illustrates application program files for plural linked application programs, in accordance with the prior art. Blocks A1, A2, A3 and A4 represent four application programs, or more particularly files created by
10 them, displayed on a screen, which is a reflection of storage of the equivalent in hard drive 9 and memory 5. Each application displays, in a toolbar 21, icons each of which is specific to the other application programs.

As described earlier in this specification, each
15 application program must both create icons relating to each of the other application programs, and also store pointers to each of the other application programs in order that when an icon related to another application program is selected from within a program, the operating application program can
20 determine which other application program to invoke.

Thus each application program in the group must contain code for, and display its own toolbar containing icons identifying the other application programs of the linked group, as well as store the corresponding pointers.

Figure 3 illustrates an implementation in illustration of the present invention. The application programs 23 do not have toolbars containing icons representing other application programs to which they are linked, or pointers to other application programs. Instead, an application program 25 is
25 devoted to a single toolbar for all application programs in the linked group. The single toolbar contains icons, each representing one of the application programs of the linked group.
30

Thus when a new or updated application program is
35 loaded into the computer for inclusion into the group, record of it need only be made in the toolbar application

program 25, and a new or updated icon included in the displayed toolbar.

In operation, when an application program is to be invoked during the operation of another application program, the icon from the single toolbar representing the program to be invoked is dragged and dropped over the file of the other application. For example, in the event a word processing program is in operation, and the user wants to send a file created by that word processing program to another person, the user would drag an icon representing a mail program from the single toolbar over the word processing file. As a result the mail program would become invoked, and open a dialog box asking the identity of the person to which the file is to be sent.

The same procedure is carried out in the event a new or updated application program has been loaded. In this case none of the group of application programs are affected by the loading of the new or updated program, and the new or updated icon from the single toolbar is dragged over the file of the application program with which its data is to be linked.

Turning now to Figure 4, examples of application programs represented in the single toolbar and operated by microprocessor 1 are shown as A1 and A2. The corresponding computer display is shown in Figure 4A. The user selects and drags an icon representing application A2 from the single toolbar over the A1 application, e.g. a word processing file, as shown by step (A). The toolbar application program 25, recognizes which application program has been dragged over which document by detecting the selected icon and the screen location where that selected icon has been dropped, and using OLE or an equivalent invokes application program A2 as shown by step (B).

It should be noted that instead of a toolbar, drop down or pop-up menus could be used. Instead of dragging and dropping, speech recognition or a touch screen can be used.

The use of the term toolbar in this specification is intended to be construed to include all such other techniques for displaying, dragging and dropping.

Further, new icons can be created for the toolbar by
5 dragging one icon on top of another, to create a series of procedures. For example, a new icon could be created constituted by a mail function followed a telephone function by dragging a mail icon in the toolbar over a telephone icon. The toolbar application program, recognizing that
10 there has been an icon over icon overlay, can present a dialog box asking whether a new icon representing the combination function should be created by the toolbar. Assuming that one has been created, dragging and dropping the new combination icon over a word processor document
15 would open up the mail application program followed by the telephone application program.

The method described is not restricted for use with computer display screen based applications, but is similarly applicable to other systems, such as control systems, or
20 user consoles of any type.

Tools for creating the display, the icons, for invoking the icons as active, and for enabling resulting operation of the application programs are well known to persons skilled in the art. For example, in the event that
25 the computer used is the Apple Macintosh, a tool that can be used is Hypercard, as described in the text "The Complete Hypercard Handbook", by Danny Goodman, copyright 1987, published by Bantam Books Inc. In the event that the computer is an IBM compatible personal computer using a
30 Windows 3.1 operating system, a tool that can be used is the programming techniques described in the text "Programming Windows 3.1", by Charles Petzold, copyright 1992, published by Microsoft Press.

A person understanding this invention may now
35 conceive of alternative structures and embodiments or variations of the above. All of those which fall within the

scope of the claims appended hereto are considered to be part of the present invention.

CLAIMS

1. A method of invoking software application programs including
 - (a) storing in a memory a toolbar application program for displaying permanently in the foreground and controlling
5 a toolbar on a computer display, separate from other application programs,
 - (b) storing in a memory other application programs which are devoid of any link to any other one of said application programs.
 - 10 (c) operating the toolbar application program and displaying a toolbar comprising at least one icon over a display of said other applications, and
 - (d) invoking a toolbar function to cause operation of at least one of said other application programs. with
15 parameters linked to an application program already in operation.
2. A method as defined in claim 1 in which said toolbar displaying step displays said toolbar transparently over at least one of an icon and a display related to a respective
5 at least one of said other application programs.
3. A method as defined in claim 1 including linking the toolbar to at least selected ones of said other application programs, displaying icons associated with each one of said selected ones of said other application programs, and
5 invoking a selected one of said other application programs by selecting one of said displayed icons from the toolbar.
4. A method as defined in claim 3 including displaying a display related to one of said other application programs, and invoking said selected one of said other application programs by dragging an icon associated with a selected one

5 of said application programs over said display related to said one of said other application programs.

5. A method as defined in claim 3 including displaying a display related to one of said other application programs, and invoking said selected one of said other application programs by speaking at least one word associated with a
5 selected one of said other application programs into a microphone.

6. A method as defined in claim 5 in which the toolbar display is acoustic, whereby no toolbar icons are visible on the screen.

7. A method as defined in claim 3 including displaying a display related to one of said other application programs, and invoking said selected one of said other application programs by touching a screen over a displayed toolbar icon
5 which displayed toolbar icon is associated with a selected one of said other application programs.

8. A method as defined in claim 1 including creating a new combined application toolbar icon by dragging an existing toolbar icon over an application icon, and invoking a sequence invoking each of the application programs
5 represented by the combined application toolbar icon over a display operated by another application program.

9. A method as defined in claim 2 in which said display related to a respective at least one of said other application programs is a window.

10. A method of invoking software application programs as claimed in claim 1, substantially as described herein with reference to Fig. 3 or Fig. 4 of the accompanying drawings.



Application No: GB 9622279.9
Claims searched: ALL

Examiner: R F King
Date of search: 23 January 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK CI (Ed.O): H4T[TBLA, TBLC, TBLM, TBLX]
Int CI (Ed.6): G06F 3/023, 3/033, 3/037
Other: ONLINE: WPI; COMPUTER

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
X	EP 0 325 885 A2 (IBM) See Fig. 7, wherein program 'MS-DOS EXECUTIVE' with a set of 'Icons', in the form of text file names enables the 'evoking' of any of the executable files listed and reference to setting up of a link list of the evoked applications whereby the applications can be 'backtracked'	1
A	US 5,422,993 (IBM) See abstract	1

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

THIS PAGE BLANK (USPTO)